Reinforcement Learning of Normative Monitoring Intensities

Jiaqi Li[1], Felipe Meneguzzi[2], Moser Fagundes[2], and Brian Logan[3]

[1] Department of Computer Science
University of Oxford
Oxford, UK
jiaqi.li@cs.ox.ac.uk
[2] School of Computer Science
Pontifical Catholic University of Rio Grande do Sul
Porto Alegre, Brazil
felipe.meneguzzi@pucrs.br, moser.fagundes@pucrs.br
[3] School of Computer Science,
University of Nottingham
Nottingham, UK
bsl@cs.nott.ac.uk

Abstract. Choosing actions within norm-regulated environments involves balancing achieving one's goals and coping with any penalties for non-compliant behaviour. This choice becomes more complicated in environments where there is uncertainty. In this paper, we address the question of choosing actions in environments where there is uncertainty regarding both the outcomes of agent actions and the intensity of monitoring for norm violations. Our technique assumes no prior knowledge of probabilities over action outcomes or the likelihood of norm violations being detected by employing reinforcement learning to discover both the dynamics of the environment and the effectiveness of the enforcer. Results indicate agents become aware of greater rewards for violations when enforcement is lax, which gradually become less attractive as the enforcement is increased.

1 Introduction

Norm-driven behaviour and monitoring have traditionally make four assumptions about the enforcement mechanism and the environment in which agents act, namely:

- the environment is fully deterministic (e.g. [11, 21, 13]);
- enforcement is either perfect or limited in known ways (e.g. "coverage" is limited [1]);
- agents are perfectly aware of all information regarding the environment and monitoring;
- agents do not change their behaviour due to changes in enforcement capability [8].

While settings based on these assumptions are a useful abstraction for theoretical work on norm-driven behaviour, when norm-driven agents are meant to either model or mimic rational decision-making behaviour in realistic environments, such as in agent-based simulation [3], they must either be relaxed or dropped entirely [12]. Consider the following example. An agent driving a car enters a city in a foreign city,[4] which has streets and traffic dynamics that are known to the agent, and the agent has a goal to drop off a passenger as close as possible to the passenger's desired destination. The agent is unaware of the meaning of the signs in this city and of the frequency with which traffic wardens patrol the streets, and must make a decision as to where to drop off the passenger, knowing that traffic may force it to stop at undesirable locations. In this kind of situation, existing approaches to norm reasoning fail to provide the agent with the means to make a decision due to a number of factors. First, although the agent is aware of the optimal way of dropping off the passenger from a movement point of view, it is unaware of exactly which spots are forbidden, and, if so, whether a sanction will be immediately applied. Second, the environment is stochastic, and some movements of the agent may be sanctioned because the environment forced the agent (by chance) to be at a certain spot. Third, if the agent makes a decision and is not sanctioned, nothing guarantees that sanctions may not be applied in the future.

In this paper, we use a reinforcement learning-based mechanism to learn normative rewards, and investigate norm enforcement mechanisms to regulate such reinforcement learning agents. Our mechanism assumes no prior knowledge of the normative state or enforcement intensity, and yields policies that are close to optimal using multiple reinforcement-learning techniques. We show the effectiveness of our approach empirically via simulations and identify key learning algorithm and parameter combinations for our scenario. The ultimate aim of our work is to allow enforcement agents to improve enforcement over time.

The remainder of the paper is organised as follows. We formalise the problem we aim to solve with our approach in Section 2 and proceed to describe our approach in Section 3, which we validate empirically using the experiments in Section 4. Finally, we compare our approach to related work in Section 5 and conclude the paper with a discussion of our contributions and directions for future work in Section 6.

2 Problem Formalisation

Norms have been widely advocated as a means of coordinating multi-agent systems and several approaches have been proposed in the literature, including state-based norms (where norms are defined in terms of states that should or should not occur), e.g., [10], and event-based norms (where norms are defined in terms of what agents should or should not do), e.g., [6, 4]. Similarly, various approaches to the implementation of norms have been proposed, including *enforce-*

[4] A city foreign to the agent's designer.

ment (where sanctions are imposed on norm-violating states and behaviours) and *regimentation* (where norm-violating states and behaviours are eliminated).

2.1 Norms and Enforcement

In this paper, we adopt essentially a state-based approach to norms and assume norms are regulated using enforcement. Each state of the environment is described in terms of a set of features $\{\varphi_1, \ldots, \varphi_n\}$, where each feature corresponds to a binary variable that must be either true or false (i.e. each feature is a propositional variable). Thus, the combination of all features (i.e. the enumeration of all possible models of $\varphi_1, \ldots, \varphi_n$) induces a state-space $\mathcal{S}$.Using such features, we define an entailment relation $\models$ over states and formulas using the standard logic connectives ($\wedge$, $\vee$, $\neg$, $\rightarrow$), so that $(s \in \mathcal{S}) \models \varphi$ means that the set of features present in s is a model of φ.

Norms specify conditions (sets of states) that either must hold (obligation) or should not hold (prohibition) when a triggering or activation condition is true. For example, parking in a no-parking zone may be prohibited between 8am and 6pm. If a norm is violated, a penalty or sanction is applied in the violation state, e.g., parking illegally may result in a fine of $100. This has some similarities to the use of 'counts as' rules in normative multi-agent programming, e.g., [5]. However we feel our approach is more intuitive in allowing the direct representation of obligation deontic modalities, rather than simply violations as in [5].

Definition 1. *A norm is a tuple* $\langle \delta, \mathcal{G}, \phi, \psi, \rho \rangle$ *where:*

- $\delta \in \{$*obligation, prohibition*$\}$ *is the deontic modality;*
- $\mathcal{G}$ *is a set of agent roles to which the norm applies;*
- ϕ *is the activation condition, which induces a set of states* $\mathcal{S}_\phi$ *such that* $\mathcal{S}_\phi = \{s \mid s \in \mathcal{S} \wedge s \models \phi\}$*;*
- ψ *is the normative condition, which induces a set of states* $\mathcal{S}_\psi$ *such that* $\mathcal{S}_\psi = \{s \mid s \in \mathcal{S}_\phi \wedge s \models \psi\}$*;*
- $\rho : \mathcal{S} \rightarrow \mathbb{R}$ *is a function that specifies the penalty for violating the norm in a given state (* $\rho(s)$ *returns the penalty to be paid in* s*).*

A norm $n = \langle \delta, \mathcal{G}, \phi, \psi, \rho \rangle$ is activated in a state $s \in S_\phi$, i.e., a state in which the activation condition ϕ of the norm holds for an agent a if the role of the agent $role(a)$ is a role to which the norm applies, $role(a) \in \mathcal{G}$. The norm is *obeyed* if the normative condition ψ holds in s (in the case of obligations) or does not hold in s (in the case of prohibitions). Otherwise the norm is *violated* in s, and the agent must pay a penalty $\rho(s)$ in s. We assume that agents are self interested, and only comply with norms if the expected penalties for non-compliance outweigh the benefits of violating a norm from the agent's perspective.

2.2 Monitoring Compliance

Norms are monitored and enforced by a normative organisation. The normative organisation is responsible for: determining when a norm is activated in a state,

whether an activated norm is obeyed or violated, and (in the case of violations), for applying the appropriate penalty.

A normative organisation monitors a set of norms $\mathcal{N}$, and (c.f. Definition 1) a state $s \in \mathcal{S}$ violates a norm $n = \langle \delta, \mathcal{G}, \phi, \psi, \rho \rangle \in \mathcal{N}$ if $\delta = \textit{prohibition}$ and $s \models \psi$, or $\delta = \textit{obligation}$ and $s \not\models \psi$. The set $\mathcal{N}_s^-$ of norms violated in state s is defined as

$$\mathcal{N}_s^- = \{\langle \delta, \mathcal{G}, \phi, \psi, \rho \rangle \in \mathcal{N} \mid \delta = \textit{prohibition} \wedge s \models \psi\} \cup \{\langle \delta, \mathcal{G}, \phi, \psi, \rho \rangle \in \mathcal{N} \mid \delta = \textit{obligation} \wedge s \not\models \psi\}$$

We assume that the probability that violations of a norm will be detected is under the control of the normative organisation. The *enforcement intensity* of the norm is a measure of the 'effort' the normative organisation is prepared to invest in detecting violations of the norm. An enforcement intensity of 1 indicates violations will be detected with probability 1, while an enforcement intensity of 0 indicates that the norm is not enforced (no violations are detected).

The enforcement intensity is modelled as a detection function $\mathcal{D}(n, t)$, which gives the detection probability of the violation of the norm $n \in \mathcal{N}$ at time step t.[5] In Fagundes [7], the detection function takes into account the current state, that is, $\mathcal{D}(n, s)$ where $n \in \mathcal{N}$ and $s \in \mathcal{S}$, but ignores the fact that in some systems the detection probabilities are not constant since they can be changed over time as part of a norm enforcement strategy.

Note that, as part of its norm enforcement strategy, a normative organisation may choose not to disclose the current enforcement intensity to the agents. This disparity in information regarding the enforcement intensity was termed *information asymmetry* in [12]. In this case, agents must determine the likelihood of norm violations being detected either by assuming the enforcement intensity to be some constant, or by trying to learn it. Criticially, given that agents cannot learn the enforcement intensity perfectly or even approximate the actual intensity without a temporal delay to observe sufficient instances of norm enforcement, it becomes possible for the normative organisation to optimise the effort spent monitoring to achieve a given level of compliance.

In the remainder of the paper we investigate norm enforcement mechanisms to regulate the behaviour of self-interested rational agents in a fully-observable stochastic environment. The mechanisms take into account not only the immediate costs and benefits of enforcing the norms with a given intensity, but also information asymmetry, and the adaptive capabilities of the agents which can change their behaviour in response to perceived changes in the norm enforcement intensity.

2.3 Example

In this section we introduce a simple Parking World scenario to illustrate the idea of variable enforcement of a norm. In the scenario, an agent drives from a

[5] In a slight abuse of notation, we shall denote by $\mathcal{D}(n)$ the detection probability of the violation of the norm $n \in \mathcal{N}$ where n is constant at all time points t

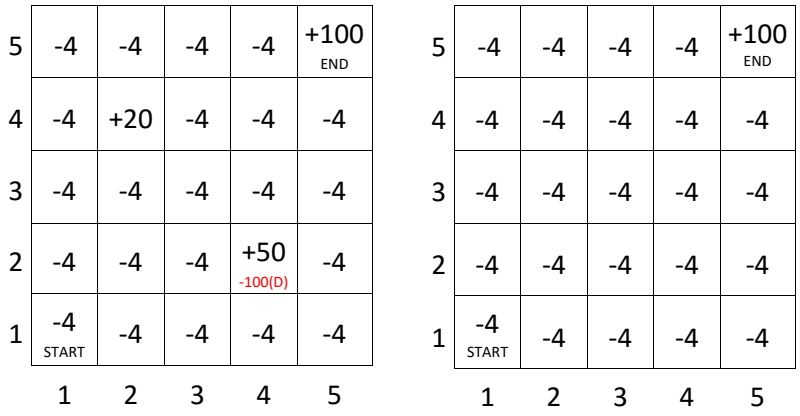
start location to a destination location (e.g., from work to home). The agent can stop on the way (e.g., to buy groceries on the way home). There are two places the agent can park: a legal parking zone, which has lower utility, but does not violate a parking norm, and an illegal parking zone, which violates the parking norm, but may have higher utility. If the violation is undetected (the parking norm is not enforced), the utility of parking illegally is higher than of parking legally; however if the norm if enforced the agent incurs a large sanction.

The Parking World is shown in Figure 1, and consists of a 5×5 grid of cells. The environment contains four distinguished cells: the *START* cell (1, 1), the *END* cell (5, 5), a "legal parking cell" (2, 4) and an "illegal parking cell" (4, 2). The agent enters the environment at the *START* cell and can move from cell to cell orthogonally and may revisit each cell apart from the *END* cell an arbitrary number of times. When the agent reaches the *END* cell the simulation stops. Visiting a parking cell counts as parking, and we assume that the agent parks at most once (legally or illegally) en route. The reward structure if the agent has not already parked is shown in Figure 1a. When visiting all cells except the *END* cell and the parking cells, the agent receives a small negative reward (penalty) of -4 (i.e., short routes between *START* and *END* have higher utility). Visiting the legal parking cell (2, 4) has a positive utility of +20. The reward for visiting the illegal parking cell (4, 2) depends on whether the normative organisation enforces the parking norm. If the norm is not enforced, the agent receives a positive reward of +50; if the norm is enforced, the agent receives a large negative reward -100 (i.e., a sanction). Visiting the *END* results in a reward of +100. The reward structure after the agent has parked at least once is shown in Figure 1b. In this case, visiting all cells except the *END* cell results in a small negative reward (penalty) of -4, and visiting the *END* results in a reward of +100. After the agent has parked once, the parking cells effectively become 'normal' cells and the parking norm is no longer enforced on the illegal parking cell (4, 2).

The scenario is designed such that the agent has to make a single decision about where to park on its way home (parking repeatedly does not increase the agent's utility). Parking legally gives a positive reward. The reward for visiting the illegal parking cell is controlled by the normative organisation. Specifically, the probability that the agent will receive a negative rather than a positive reward for parking illegally is determined by the enforcement intensity. Critically, the agent has limited information about the enforcement intensity of the parking norm. However the agent can attempt to learn the enforcement intensity over multiple trials, and we discuss this in the next section.

3 Reinforcement Learning in Normative Organisations

The process of reinforcement learning (RL) can be described as follows: an RL agent first obtains the initial state of the environment and then selects and executes an action. The environment then responds with a numerical reward and a new state. The agent makes its second move based on the reward it received



(a) Rewards before parking (b) Rewards after parking

Fig. 1: Two Layer Parking World

in the first step and the new state. This process repeats until the agent reaches the end state or it cannot proceed any further. For example,

$$s_0 \xrightarrow{a_0 \quad r_1} s_1 \xrightarrow{a_1 \quad r_2} s_2 \ ... \ s_{n-1} \xrightarrow{a_{n-1} \quad r_n} s_n$$

where s_n is state n, a_n is an action made by agent at state n, $r_{n+1} = R(s_{n+1})$ is the reward given by the environment for reaching state $n+1$.

RL can be formulated as Markov Decision Process (MDP). An MDP is a five-tuple, $MDP =< S, A, P(s'|s,a), R >$, where

- S is a set of possible states. For all states in discrete time steps, $s_t \in S$.
- A is a set of possible actions, where for all actions a possible in a given state, $a \in A$.
- $P(s'|s,a)$ is the probability of moving from state s to s' when executing action a, such that $\sum_{s'} P(s'|s,a) = 1$ and $P(s'|s,a) \geq 0$.
- R is the reward function, mapping from states to reward values. $R : S \mapsto \mathbb{R}$.
- γ is the discount factor, $0 \leq \gamma \leq 1$. The discount factor determines the importance of future rewards.

The goal of agent is to maximise its total reward, $\Sigma_{i=0}^{n} R(s_i)$, by computing a control policy. The policy is a function, π, that maps from each possible state of the environment to an action.

$$\pi : S \to A$$

The optimal policy is obtained by learning a value function, V, that maps each state (or state-action pair for Q-learning) to a numeric value, indicating expected total reward following that state. The optimal value function, V^*, is defined as

$$V^*(s) = R(s) + \max_{a \in A} \gamma \sum_{s' \in S} P(s'|s,a) V^*(s')$$

Therefore, the optimal policy, π^*, can be defined as the best action a such that future expected total reward is maximised in state s.

$$\pi^*(s) = \arg\max_{a \in A} \sum_{s' \in S} P(s'|s,a)V^*(s')$$

We assume that the agents do not know exactly what the rewards are for states where norms apply, nor exactly which states are affected by norms (and thus, the probability of being sanctioned). We have therefore developed an approach that can use any model-free reinforcement learning mechanism, and have implemented the two most common reinforcement learning algorithms, namely SARSA [15] and Q-Learning [20].

A key problem in RL, is balancing exploration and exploitation. For each step, the agent needs to decide whether to follow the best action given by its learned policy (exploitation) or randomly pick an action (exploration). It's obvious that we cannot do exploration all the time, which means that agent makes no use of the learned knowledge about the environment. On the other hand, exploitation fails to discover potential better actions. In our approach, we use an epsilon-greedy strategy, which chooses an exploitation action in most cases; however with probability ϵ, the agent chooses a random action. This guarantees that eventually all states are visited after an infinite number of runs. If a^* is the optimal action given by agent's policy, the probability of choosing an action $a \in A$ using an epsilon-greedy strategy is:

$$P(a) \leftarrow \begin{cases} 1 - \epsilon + \frac{\epsilon}{|A|}, & \text{if } a = a^* \\ \frac{\epsilon}{|A|}, & \text{if } a \neq a^* \end{cases}$$

In practice, ϵ should be large enough to help the agent interact with the environment and learn quickly, and small enough to maximise the total rewards. A value of 0.1 is often used in the literature.

The basic reinforcement learning algorithm is shown in Algorithm 1. Each trial t represents a full execution of the agent starting from the initial state to the end state. The current state s and the next state s' are initialised to the initial state s_0. For each step in a trial, if the agent is not already in the end state, we calculate all applicable states of the agent given its current state. The exploration-exploitation strategy then decides whether the agent chooses exploration (selects a random action), or exploitation (an action given by agent's policy). The next state s' is given by executing the action in a non-deterministic environment. The optimal next state s^* is the state followed by agent's policy without consideration of environment (Algorithm 3). The mechanism of assigning the reward of each step r is given Section 2.3 and in detail in Algorithm 2. Finally, we update the utilities of states (or state-action pairs) using either SARSA or Q-learning algorithms (Algorithms 4 and 5) (depending on the experimental setup, see Section 4).

SARSA and Q-Learning are temporal difference approaches to learning the optimal policy. The core of the two algorithms are update equations for the

Algorithm 1: Reinforcement learning for the Parking World

```
1  foreach t ∈ totalTrials do
2      s ← s' ← s_0
3      while ¬isTerminal(s) do
4          A ← applicableActions(s)
5          if isExploration(t) then
6              a ← random a ∈ A
7          else
8              a ← π(s)
9          s' ← execute(a)
10         s* ← getOptimalState(s)
11         r ← getReward(s)
12         V(s) ← Update(s, s', s*, α, γ)
13         s ← s'
       /* Update final state                                    */
14     r ← getReward(s)
15     err ← r + 0 − V(s)
16     V(s) ← V(s) + α ∗ err
17     reset
```

Algorithm 2: getReward(state)

```
1  if s = illegalState then
2      if isDetected(s) then
3          r ← penaltyOfIllegalParking;
4  else
5      r ← R(s);
   /* Once the illegal cell or the legal cell is visited, they become
      'normal' cells and norms are not enforced                 */
6  if s = illegalState or s = legalState then
7      illegalState.reward ← defaultReward;
8      legalState.reward ← defaultReward;
9      enforcementIntensity ← 0
10 return r
```

Algorithm 3: getOptimalState(state)

```
1  foreach a ∈ applicableAction(s) do
2      s' ← execute(a);
3      if V(s') > V(s*) then
4          s* ← s'
5  return s*
```

value function, which are given in Algorithm 4 (for SARSA) and Algorithm 5 for (Q-learning). α is the learning rate that controls the amount of difference that contributes to the update of the value of state s.

Algorithm 4: SARSA: Update(s, s', s^*, α, γ)

1 **return** $V(s) + \alpha \cdot [r + \gamma \cdot V(s') - V(s)]$

Algorithm 5: Q-learning: Update(s, s', s^*, α, γ)

1 **return** $V(s) + \alpha \cdot [r + \gamma \cdot V(s^*) - V(s)]$

The SARSA and Q-learning algorithms have distinct characteristics [16, p. 844] when exploration takes place. In this context, Q-learning is more flexible in the sense that it is able to converge towards an optimal policy even if the initial policy is random or very low quality, since its update rule always takes the best Q-value backed up so far. Conversely, SARSA is more realistic in that its update rule always uses the actual values obtained in each learning episode (and thus has less bias towards optimistic assessments). This has important implications for the results we might expect from these algorithms in our approach. Namely, we expect Q-learning to perform better when the penalties for violation are high, resulting in a norm-compliant policy substantially different than a norm-ignoring policy, whereas we expect SARSA to perform better when the enforcer agent changes enforcement more often.

4 Experiments

We carried out two experiments using the scenario described in Section 2 to study the behaviour of the SARSA and Q-learning agents:

1. under different fixed enforcement intensities; and
2. under variable enforcement intensities.

In both experiments, we used the following parameter values (the meanings are explained Section 3):

- the ϵ-greedy strategy has $\epsilon = 0.1$;
- the discount factor, γ, is set to 0.9; and
- the learning rate, α, is set to 0.01 for the first 100,100 trials to help the agents learn efficiently in the earlier trials.

$$\alpha \leftarrow \begin{cases} \frac{1}{c-100000}, & \text{if } c \geq 100,100 \\ 0.01, & \text{if } c < 100,100 \end{cases} \quad (1)$$

where c is the number of times that a cell has been visited.

4.1 Fixed Enforcement Intensities

We know that if the enforcement intensity is high, choosing a path that goes through the legal parking cell gives a higher total reward. On the other hand, choosing a path through the illegal cell is better if the enforcement intensity is low. Therefore, there exists a critical value of enforcement intensity where the agent switches its preference from a path through the legal cell to a path through the illegal cell or vice versa. The purpose of the first experiment is to find this critical value.

In the first experiment, we varied the enforcement intensity from 0 to 1 in steps of 0.1. Having identified the critical range of values, we performed a further set of experiments using a step size of 0.01. For each experiment the agent was run 10 times with 1 million episodes per run to obtain an average utility.

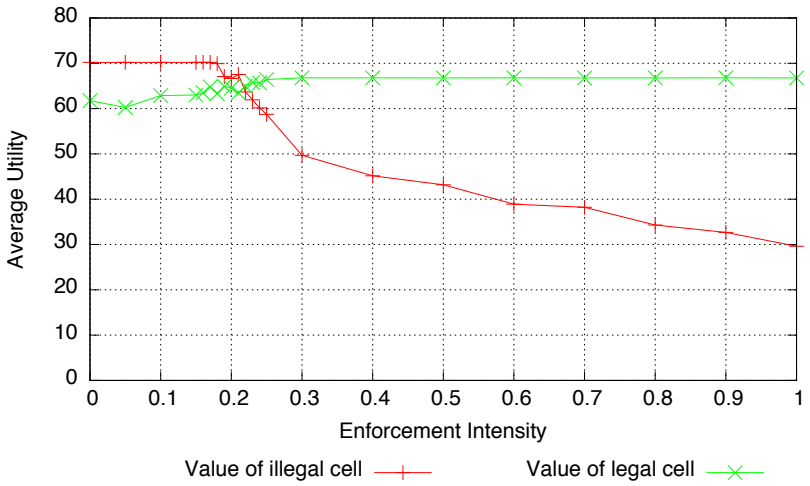


Fig. 2: Learned utilities for differing enforcement intensity (SARSA)

The results for the SARSA agent are shown in Figure 2. As can be seen, the critical enforcement intensity is around 0.22. Moreover, Figure 2 shows that when the enforcement intensity is greater than 0.3, further increases in enforcement intensity have no significant effect on the utility of the legal cell. Similarly, when the intensity is below 0.18, decreasing the enforcement intensity has very little effect on the utility of illegal cell.

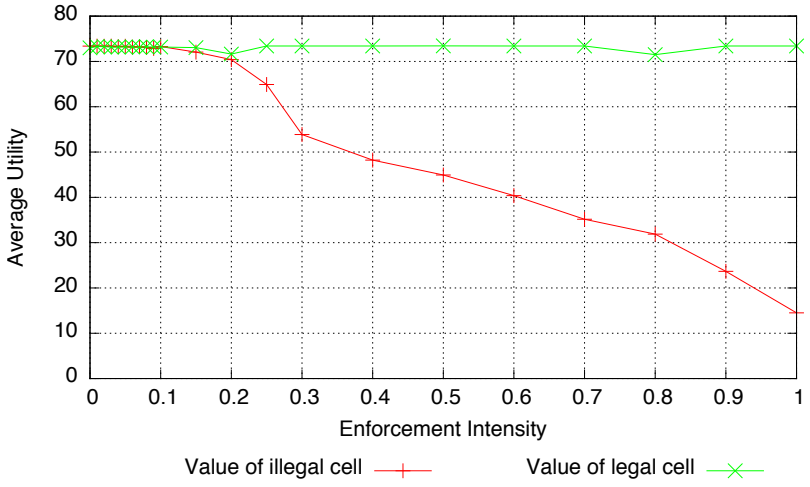


Fig. 3: Learned utilities for differing enforcement intensity (Q-learning)

The results for the Q-learning agent are shown in Figure 3. In this case, the utilities of the legal and illegal cells are very close when the enforcement intensity is lower than 0.15. While the utility of the legal cell is stable when the enforcement intensity increases, the utility of the illegal cell decreases.

4.2 Variable Enforcement Intensities

The second experiment was designed to show how the SARSA and Q-learning agents behave when the enforcement intensity changes during a single run. This experiment was divided into two phases. In the first phase, the agent was trained with an enforcement intensity of 0 until its learned utilities for the legal and illegal cells converged. The agent was then evaluated 1,000 times using this learned policy (the policy was kept fixed during the evaluation period). The agent then entered the second phase with the policy it learned in the first phase. In this phase, the enforcement intensity was changed to 1 and the agent was trained 1,000 times followed by 1,000 runs for evaluation (again the policy learned in the second phase was kept fixed during the evaluation period). We ran this experiment 10 times and took the average total reward of each episode. Since the length of the training period in the first phase varied from run to run, we do not report the data collected in this period in our results.

The total rewards for the first 1,000 runs collected during the evaluation of the SARSA agent trained under an enforcement intensity 0 are shown in Figure 4. In this period, the agent chooses the illegal path as no sanctions are applied, resulting very high total rewards (about 110 on average). At the 1,001st episode, the intensity changes to 1, resulting in all illegal parking being punished. As a consequence, the total reward drops immediately, because the agent continues

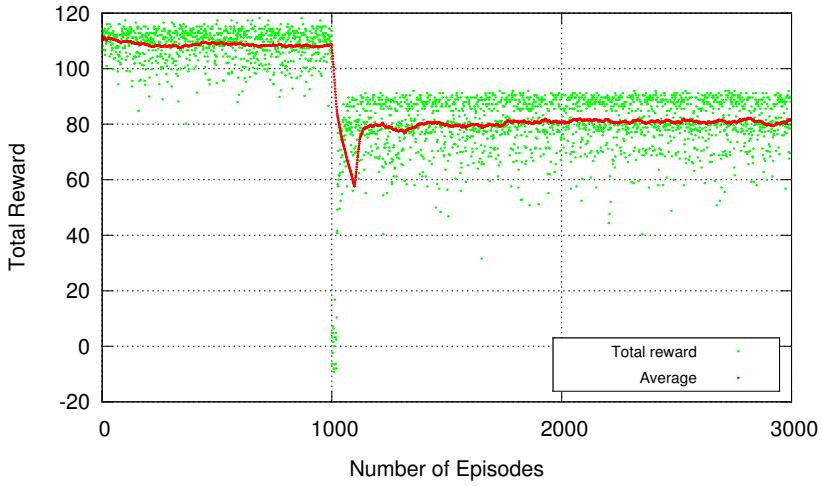


Fig. 4: Total rewards received by the SARSA agent for enforcement intensities 0 and 1. The green dots are the average total reward of 10 runs and the red dots are the averages of 100 recent green dots.

to follow a policy that believes an illegal path is better, which is no longer correct. However, the SARSA agent is able to adapt to this change very quickly. After a few episodes with very low total rewards, its policy is updated to a legal path. As we can see from the figure, the average total rewards after the change in enforcement intensity is about 80. In addition to this main result, we also observe that the average total reward of an illegal path is 30 units higher than an legal one. This is expected and is exactly what we defined in the scenario, where the rewards of the legal cell and illegal cell are 20 and 50 respectively.

However, the results of the first evaluation period for the Q-learning agent (first 1,000 episodes in Figure 5) fluctuate, as it has difficulty deciding between a legal path or an illegal one, i.e., the utilities of legal and illegal parking cells are very close when the enforcement intensity is 0. When the intensity is increased to 1 after the first 1,000 episodes, the agent also quickly learned a new policy which gives similar total rewards as the SARSA agent. At the moment we are investigating the causes of this behaviour, and our future work aims to use different scenarios to replicate it and hopefully explain under what conditions this shift in utility happens.

5 Related Work

Our work builds upon the basic model of Fagundes et al. [9]. While the NMDPs in that work are slightly more expressive in allowing penalties in the form of enforced transitions, the basic assumption of constant enforcement intensity

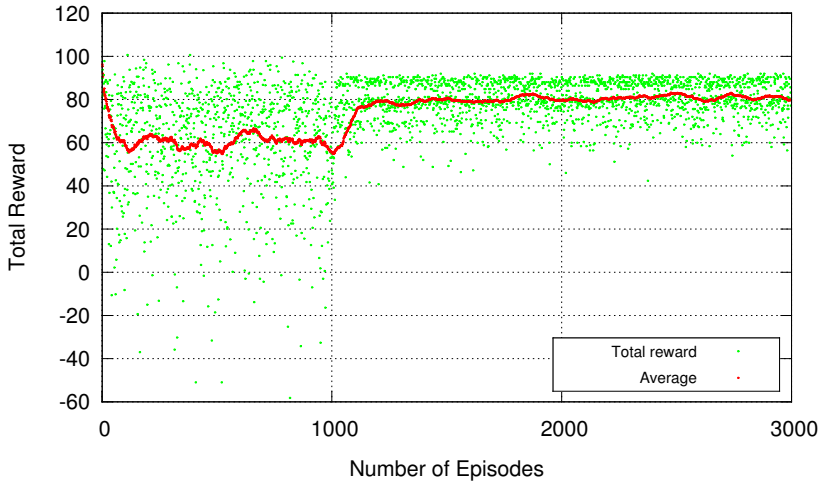


Fig. 5: Total rewards received by Q agent under different enforcement intensity. The green dots are the average total reward of 10 runs and the red dots are the averages of 100 recent green dots.

throughout an agent's lifetime in [9] precludes the kind of learning mechanism and normative organisation adaptation we define in this paper.

Our work is also related to work on norm identification. Norm identification techniques have mostly been developed in deterministic environments, with a focus on identifying the actual norms present in a normative MAS rather than in detecting the enforcement intensity of the norms [17]. Savarimuthu et al. [18, 19] propose learning-based norm identification mechanisms to identify conditional norms. This work differs from ours in two fundamental respects: first, it assumes the norms are not known (and the task is discovering them and their conditions), and second, the environment is deterministic. In addition, their use of learning techniques focuses on data-mining techniques to be used in environment interaction histories, whereas our work is based on the use of reinforcement learning by an agent acting in the environment. Thus, whereas Savarimuthu's agents learn norms by observation, ours learn the enforcement intensity of (known) norms by acting on the environment.

Morales at al. [14] have proposed a mechanism for the automated synthesis of norms that ensures norms are conflict free and achieve certain coordination properties. In contrast, we assume the set of norms is static, and the synthesis approach proposed by Morales et al. does not consider the possibility of imperfect or variable enforcement. We believe the combination of norm synthesis approaches and a variable enforcement mechanism are a promising avenue of future work.

6 Conclusion

Our experiments show that reinforcement learning agents can: 1) learn different policies to maximise their total rewards under different unknown norm enforcement intensities in a non-deterministic environment; and 2) adapt to a change of enforcement intensity very quickly so as to obtain maximum total reward under the new enforcement intensity.

There are several directions for future work. The behaviour of the Q-learning agent under low enforcement intensities requires further investigation to explain why the agent is unable to choose between the legal and illegal parking cells. In addition, we would like to include enforced transitions in our learning framework, e.g., instead of a penalty after violation, the agent is returned to the initial state. Finally, we plan to explore the behaviour of the normative organisation, i.e., how the normative organisation can maximise its utility by changing enforcement intensities given the agent's policies.

References

1. Alechina, N., Dastani, M., and Logan, B. Norm approximation for imperfect monitors. In *Proceedings of the International conference on Autonomous Agents and Multi-Agent Systems, AAMAS* (2014), pp. 117–124.
2. Bazzan, A. L. C., Huhns, M. N., Lomuscio, A., and Scerri, P., Eds. *International conference on Autonomous Agents and Multi-Agent Systems, AAMAS '14, Paris, France, May 5-9, 2014* (2014), IFAAMAS/ACM.
3. Beheshti, R., and Sukthankar, G. A normative agent-based model for predicting smoking cessation trends. In *Proceedings of the International conference on Autonomous Agents and Multi-Agent Systems* (2014), pp. 557–564.
4. Cliffe, O., De Vos, M., and Padget, J. Specifying and reasoning about multiple institutions. In *In the AAMAS06 Workshop on Coordination, Organization, Institutions and Norms in agent systems (COIN-2006* (2006).
5. Dastani, M., Meyer, J.-J. C., and Grossi, D. A logic for normative multi-agent programs. *J. Log. Comput. 23*, 2 (2013), 335–354.
6. Esteva, M., de la Cruz, D., and Sierra, C. ISLANDER: an electronic institutions editor. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems* (New York, NY, USA, 2002), AAMAS '02, ACM, pp. 1045–1052.
7. Fagundes, M. S. *Sequential Decision Making in Normative Environments*. PhD thesis, Universidad Rey Juan Carlos, 2012.
8. Fagundes, M. S., Billhardt, H., and Ossowski, S. Reasoning about norm compliance with rational agents. In *ECAI* (2010), H. Coelho, R. Studer, and M. Wooldridge, Eds., vol. 215 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, pp. 1027–1028.
9. Fagundes, M. S., Ossowski, S., Luck, M., and Miles, S. Using normative markov decision processes for evaluating electronic contracts. *AI Communications 25*, 1 (2012), 1–17.
10. Hübner, J. F., Sichman, J. S., and Boissier, O. Developing organised multi-agent systems using the $\mathcal{M}$OISE$^+$ model: programming issues at the system and agent levels. *Int. J. Agent-Oriented Softw. Eng. 1*, 3/4 (2007), 370–395.

11. Kollingbaum, M. J., and Norman, T. J. Norm adoption and consistency in the noa agent architecture. In *Programming Multi-Agent Systems* (2003), M. Dastani, J. Dix, and A. E. Fallah-Seghrouchni, Eds., vol. 3067 of *LNCS*, Springer, pp. 169–186.
12. Meneguzzi, F., Logan, B., and Fagundes, M. S. Norm monitoring with asymmetric information. In Bazzan et al. [2], pp. 1523–1524.
13. Meneguzzi, F., and Luck, M. Norm-based behaviour modification in BDI agents. In *Proceedings of the Eighth International Conference on Autonomous Agents and Multiagent Systems* (2009), pp. 177–184.
14. Morales, J., Lopez-Sanchez, M., Rodriguez-Aguilar, J. A., Wooldridge, M., and Vasconcelos, W. Automated synthesis of normative systems. In *Proceedings of the 2013 international conference on Autonomous agents and multi-agent systems* (Richland, SC, 2013), AAMAS '13, International Foundation for Autonomous Agents and Multiagent Systems, pp. 483–490.
15. Rummery, G. A., and Niranjan, M. On-line q-learning using connectionist systems. Tech. Rep. TR 166, Cambridge University Engineering Department, 1994.
16. Russell, S. J., and Norvig, P. *Artificial Intelligence - A Modern Approach*, 3rd ed. Pearson Education, 2010.
17. Savarimuthu, B. T. R., and Cranefield, S. Norm creation, spreading and emergence: A survey of simulation models of norms in multi-agent systems. *Multiagent and Grid Systems 7*, 1 (2011), 21–54.
18. Savarimuthu, B. T. R., Cranefield, S., Purvis, M. A., and Purvis, M. K. Obligation norm identification in agent societies. *Journal of Artificial Societies and Social Simulation 13*, 4 (2010).
19. Savarimuthu, B. T. R., Cranefield, S., Purvis, M. A., and Purvis, M. K. Identifying conditional norms in multi-agent societies. In *Coordination, Organizations, Institutions, and Norms in Agent Systems VI*, M. De Vos, N. Fornara, J. Pitt, and G. Vouros, Eds., vol. 6541 of *Lecture Notes in Artificial Intelligence*. Springer, Berlin / Heidelberg, 2011, pp. 285–302.
20. Watkins, C. J. C. H. *Learning from Delayed Rewards*. PhD thesis, King's College Cambridge, 1989.
21. Yan-bin, P., Gao, J., Ai, J.-Q., Wang, C.-H., and Hang, G. An extended agent BDI model with norms, policies and contracts. In *4th International Conference on Wireless Communications, Networking and Mobile Computing* (Oct. 2008), pp. 1–4.